Final Weather app.

Overview

My project - "Weather App" is made to use the OpenWeatherMap API to retrieve weather data for a given city, display it on a webpage, and save it in a MySQL database and if cache is available go directly to user’s webpage then save it in a MySQL database as well. The advantages and disadvantages of my application's architecture will be examined below.

Advantages

Database Integration: Reliable weather data storage for every city is ensured via a smooth integration with a MySQL database, which makes use of SQL queries for data consistency.

Asynchronous requests: To ensure a seamless user experience without the UI stalling, JavaScript functions like “checkweather()” employ asynchronous calls to get data from the OpenWeatherMap API.

Error Handling: To enable quick problem detection and resolution, my code includes error-handling techniques for managing SQL query failures and checking for database connection faults.

Cache: Although storing weather data in the client's local storage presents security and privacy problems, the app will still function by retrieving information from there and displaying it to the user in case that the user's network is unstable.

Disadvantages

Security: Because user input is entered into SQL queries directly without being properly sanitized, it have a risk of leaking SQL data.

Error handling. Although I have added some error handling but it is not relevant for users. For example, if API requests fail due to network issues or incorrect input, the user is not given clear error messages.  
  
User interface improvements. To let users know how their queries are progressing, the user interface does not include any feedback methods such as loading indicators or error warnings. I will definitely address this issue in the future, which will improve user satisfaction and usability.  
  
  
  
  
  
  
  
  
  
  
In summary   
  
Although the "Weather App" shows some basic functionality, in the future fixing the found flaws will improve its security, dependability, and overall user experience.